January 11, 2002
Ref. No.: EOS/ETS-011102-012

National Aeronautics and
Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

Attention: Mr. Willie Fuller
Code 581
Building 32, Room N212D

Subject: Contract No.: NAS9-98100
CSOC SODA Task Order Number GC36
EOSDIS Test System (ETS) Multimode Portable Simulator for Aura
(MPS/Aura) Delivery of the Release 3.0 Software

Dear Mr. Fuller:

We are pleased to deliver Release 3.0 of the ETS Multimode Portable Simulator (MPS) for the Aura spacecraft (MPS/Aura). This is the fourth major delivery of MPS/Aura using our Scalable, Integrated Multimission Simulation Suite (SIMSS) infrastructure and architecture with extensions for the EOS Aura spacecraft.

Release 3.0 was previously delivered as a Beta version while awaiting an Aura Project Data Base (PDB) to facilitate final testing. When received, the PDB did not contain any useable HIRDLS commands. HIRDLS commands use a Function Code for command identification. For that reason it may be necessary to deliver a patch release to repair the command recognition code. However the major functionality in MPS/Aura has been tested and found to be working correctly.

The major capabilities in this release consist of adoption of the Aura PDB schema, use of the Red/Yellow Limits PDB file to refine initial telemetry values, and recognition of commands via spacecraft-specific information in the PDB. A complete list of the simulator capabilities is included in Attachment A.

A hard copy of the MPS/Aura User's Guide for Release 3.0 will be made available to the users in the near future. In addition, a soft copy of the User's Guide will be placed on the ETS Documentation web site at URL http://esdis-it.gsfc.nasa.gov/ETS/etsdoc.html.

This delivery package contains 12 attachments as listed below. A completed Mission Systems Configuration Management (MSCM) form is included in Attachment L. If you have any questions concerning this delivery, please call me at 301-805-3420.

Sincerely yours,

Dave Green,
CSOC ETS Task Leader

EOSDIS Test System (ETS) Multimode Portable Simulator for Aura (MPS/Aura)
Delivery of the Release 3.0 Software
January 11, 2002, Ref. No.: EOS/ETS-011102-012

Delivery Package Reviewed and Approved by:

Charles Reed, Jr.
CSOC Quality Assurance

Estelle Noone
CSOC ETS Customer Service Representative

The following attachments contain the details of the MPS software delivery.

Attachment A - describes the delivery contents for this release
Attachment B - describes the operational changes
Attachment C - contains the instructions to build and install the software
Attachment D - contains any special operating instructions
Attachment E - contains a list of the resolved DRs
Attachment F - contains a list of the unresolved DRs
Attachment G - contains the matrix of requirements addressed by this release
Attachment H - contains the known system limitations
Attachment I - contains the release history summary matrix
Attachment J - contains a listing of the delivery media contents
Attachment K - contains documentation references
Attachment L - contains the Mission Systems Configuration Management (MSCM) form

Distribution:

NASA

Caulfield, M.
Johns, A.
Kelly, A.
Krimchansky, A.
Lehtonen, K.
Ondrus, P.
Perkins, D.
ESDIS Library

Honeywell

Luo, C.
Mateik, D.

CRC

Lavery, K.

Lockheed Martin

Cordier, G.
Garner, G.
Johnson, T.

General-Dynamics

Patterer, R.

Raytheon

Gradishar, G.
Johnson, W.
Klem, K.
Ruland, V.

Averstar

Messerly, R.
Sexton, D.
Winters, R.

CSC

Burrows, P.
Carlson, J.
Green, D.
Noone, E.
Quintin, E.
Reed, C.
Ramey, D.
Shurie, E.
Task File

Attachment A – Description of Delivery Contents

The MPS/Aura Release 3.0 consists of custom software executables that are being delivered on one CD-ROM.   Two copies are being provided.

Enterprise Oracle and Oracle Programmer for Windows NT are necessary for operation of this release. These products were provided with a previous release. The license to use Oracle belongs to the CSOC contract. Therefore, Oracle and Oracle Programmer may be installed only on CSOC computers.

A soft copy of this MPS/Aura Release 3.0 delivery letter and set of attachments is also being delivered. The attachments have been formatted on a 3.5" IBM PC diskette utilizing the MS WORD word processing tool.

Attachment B – Summary of Operational Changes

Operational Capabilities of MPS/Aura Release 3.0

New or modified capabilities with this release are noted in **Bold**.

Telemetry:

- Transmit telemetry in IP or Serial (clock/data) mode
- Pack telemetry packets and CLCWs into CADUs when in Serial mode
- Generate one stream of CADUs when in Serial mode
- Generate one stream of telemetry formatted as EDUs when in IP mode
- Start or stop one telemetry stream
- Ingest the PDB files
- Generate telemetry packets from information contained in the PDB
- Maintain telemetry nodes from information contained in the PDB
- Populate telemetry packets with data values from information contained in the PDB
- Generate correct secondary headers for SC, GIRD, and SUROM-TIE (no secondary header) telemetry packets using information from the PDB
- Generate instrument telemetry packets using secondary key information from the PDB
- Display EDU data when in IP mode
- Display CADU data when in Serial mode
- Set values into telemetry points by mnemonic
- Display telemetry node values by mnemonic
- Convert telemetry values to Engineering Units (EU) for display using information from the PDB
- Accept operator-entered telemetry values in EU and convert to Raw Counts for inclusion in telemetry packets
- Reset packet count for the telemetry stream
- Static packet data can be overwritten (by byte location) and by modification of telemetry mnemonic
- Incrementing packet sequence counters per APID
- Generation of individual APIDs can be inhibited
- Telemetry logs will be created (viewable by offline utility)
- Packet Headers and Packet Data are updated
- Packet data can be shown in hexadecimal or octal format and addressed in hexadecimal or decimal form
- Packet Sequence Counters can be reset
- Packet Sequence Counters can be modified
- Packet Version field can be modified
- Packet APID field can be modified
- Packet Type field can be modified
- Packet Secondary Header Flag field can be modified
- Packet Length field can be modified

- CCSDS Unsegmented TimeCode (CUC) can be modified
- Packet rate may be controlled
- CLCW transmitted via EDUs when in IP mode
- IP packets are transmitted with variable lengths
- CLCW can be overridden by the operator
- Transmission of CLCW can be inhibited when in IP mode
- Scenario file (script) capability to set telemetry nodes and buffers
- Set telemetry data values in response to spacecraft commands received (end-item verification)
- Set initial telemetry data values at initialization
- Allow simultaneous display and set of multiple telemetry container items via GUI screens
- Simulate spacecraft memory dumps
- Use the PDB telemetry state text file to locate end-item verifier values
- Maintain and update telemetry data values in APID 1000
- Telemetry parameters may be set and viewed by Parameter ID
- CLCW Transmit Start and Stop is coupled to H/K Telemetry Start and Stop
- Telemetry values may be set using simple expressions
- Telemetry values may be set using trigonometric expressions
- Telemetry values may be set using Boolean expressions
- Telemetry values may be set to other telemetry mnemonic values
- Telemetry values may be saved in intermediate variables for later use
- TES Segmented Packets are emulated
- CLCW Transmit rate may be set by the operator
- Telemetry data values are validated for fit into packet space
- Current enable status and transmit rate for all APIDs is viewable via status display
- vcProcessor module discards VC63 VCDUs when creating files for playback
- **The PDB Red/Yellow Limits file is used to refine initial telemetry values.**
- **Signed telemetry data values are validated as one's and two's complement integers upon user input, as appropriate.**
- **Displays of telemetry and command container item names may be saved and restored.**
- **The VCDU Sequence Counter field occupies 32 bits in APID 1000.**

Command:

- Ingest command-related PDB files
- Identify commands using information from the PDB
- Display event messages with command mnemonics and submnemonics
- Set telemetry points in response to commands received (end-item verification) using information from the PDB
- Recognize spacecraft Command Loads
- Display Command Load data

- Copy Command Load data to a Memory Dump buffer
- Inhibit the Command Load data copy facility via operator directive
- Validate checksums of received Command Loads
- Ingest type AD, BC, and BD commands
- Display Total CLTUs count
- Reset Total CLTUs count
- Display Rejected CLTUs count
- Reset Rejected CLTUs count
- Display Instrument commands count
- Reset Instrument commands count
- Display Spacecraft commands count
- Reset Spacecraft commands count
- Display BC commands count
- Reset BC commands count
- Display BD commands count
- Display current Spacecraft CLCW
- Update Spacecraft and instrument CLCW
- Display current Instrument CLCW
- Validate commands based on individual, all, or none of the following validation criteria: CLTU Start and Tail Sequences, BCH Error Code, Transfer Frame Header Fields, FARM (Valid Frame Sequence), User Command Packet Header
- Generate event messages based on ingest
- Log raw commands (viewable by offline utility)
- Display raw command in hexadecimal or octal format addressed in either hexadecimal or decimal fashion
- Display command packet headers for instrument commands
- Display command packet headers for spacecraft commands
- Update command accepted and rejected counters in telemetry
- Command submnemonics are saved in container items and may be viewed after command receipt
- Expected Spacecraft ID changed to CC Hex
- **TES and OMI segmented commands are recognized.**
- **The Function Code is used to identify HIRDLS commands.**
- **The two’s complement checksum of instrument commands is validated.**

Time:

- Maintain and update SC time (GIRD)
- Maintain and update GMT time
- Synchronize SC and GMT times

General:

- Control all simulator module functions via scenario scripts
- Selection of scenario scripts may be via operator type-in or via a file selection browse window
- Start scenario scripts in response to commands received
- Start a scenario script from a scenario script
- Execute multiple scenario scripts simultaneously
- Provide operator control of multiple scenario scripts started by the operator
- Save the last 10 operator directives
- Allow editing of saved operator directives before re-execution
- EDOS Service Header (ESH) fields may be viewed
- ESH field contents may be modified by the operator
- Validation of Command Data Block (CDB) header fields of commands received
- Modification of expected values of CDB header fields
- All viewable buffers may be displayed
- Addition, deletion, and modification of command end-item verifiers via SQL scripts
- Logs of commands received or telemetry transmitted may be retransmitted via IP output or Serial output
- Expected Spacecraft ID may be modified in EOSGS module
- CLCW ESH field contents may be modified by the operator
- Event messages to the screen may be inhibited or enabled by severity (color)
- Scenario scripts may contain IF-then-ELSE-ENDIF and WHILE-ENDWHILE conditional execution directives
- The Scenario module may interface with multiple modules
- Intermediate variables A – Z permit saving values as real numbers
- Intermediate variables Aq – Zq permit saving values as long integers
- **The Aura PDB DFCD schema has been adopted**
- **The Load Database window can accept a 32 character version designator**
- **The Serial Output module can accept directives from the operator or via a scenario script.**
- **The Event Message window has been separated from the project window and has been made resizable.**

Attachment C – Installation Instructions for MPS/Aura Release 3.0

This attachment contains the instructions for installing the PDB files and the MPS/Aura Release 3.0 Server and Client. The information presented in this attachment is divided into three major sections. The first section contains abbreviated installation instructions, the second contains a summary of the installation changes, and the third section contains detailed instructions for performing initial and subsequent installations.

The information presented in this attachment has been checked for accuracy by the independent test team.

IMPORTANT

Once the Oracle database has been set to the Aura schema definition it is not possible to run Release 6.4 of the MPS/Aqua simulator using that database. It is recommended that only one PC be updated to the new schema. Instructions for accessing a PDB across a network are given in Attachment D.

C-1: Abbreviated Installation Instructions

These instructions are intended for the experienced user performing an installation on a PC where Oracle has already been installed.

1. Install the MPS/Aura Release 3 Client software by executing the **Setup.exe** program in the Client folder of the CD.
2. Install the MPS/Aura Release 3 Server software by executing the **Setup.exe** program in the Server folder of the CD.
3. If the new database scripts were not previously installed, install them by executing the **Database.bat** program in the Database folder of the CD.
4. If not previously done, create a folder under **D:\mps_pdb\AuraPDBs** to hold the Aura PDB source files. Copy the Aura PDB source files into this new folder. Twelve files are needed. See the list in Paragraph C-3.5 for the files to be copied.

5. If the Aura-compatible database tables were not previously created, as during installation of either MPS/Aura Release 3.0 Beta or MPS/Aqua Release 6.5, navigate to the **D:\mps_pdb\GenericScripts** folder and execute the **Build_PDB_Tables.bat** batch program. This creates the Aura-format tables within Oracle. **ONLY PERFORM THIS STEP ONCE!**

6. Navigate to the **D:\mps_pdb\AuraScripts** folder and execute the **Load_Next_Aura_PDB.bat** batch program. This will ingest the Aura PDB into Oracle. There are two important points to follow:

    - When navigating to the folder containing the PDB source files, double-click each folder name except the last. The folder containing the PDB files is selected, not opened.
    - Spaces are not permitted when entering the PDB version designator.

7. When initializing the MPS/Aura simulator for the first time, any IP-mode projects needed must be built and saved.

C-2: Summary of changes

Several changes had to be made to accommodate the Aura PDB schema changes while allowing Aura and Aqua PDBs to reside in the same Oracle repository. These changes are:

- The PDB root folder is now **D:\mps_pdb.** Sub-folder details are given in Paragraph C-3.4.
- The Oracle ingest scripts now attach a spacecraft name to every record. The field name is **sc_name**. In addition to the version designator, all SQL queries must now include either “sc_name = ‘AURA’” or “sc_name = ‘PM1’”. The single quotes are required syntax. The MPS simulators automatically append the correct spacecraft name when accessing the Oracle database.
- PM1 was selected as a spacecraft name vs. Aqua to minimize the chance of errors.
- The last Aura PDB loaded AND the last PM1 PDB loaded will each be the default for that spacecraft name within Oracle.
- The version designator has been increased to 32 characters. **Spaces are not allowed. All alphabetic characters will be translated to upper case.**
- The batch file to create the Oracle tables has been renamed to **Build_PDB_Tables.bat**
- There is one batch file to ingest all Aura PDBs. It is named **Load_Next_Aura_PDB.bat**
- There is one batch file to ingest all Aqua (PM1) PDBs. It is named **Load_Next_PM1_PDB.bat**

C-3: Detailed Installation Instructions

This is the complete procedure for performing an initial or subsequent installation of the MPS/Aura simulator Release 3.0, and associated software, data files, and COTS programs on a PC. There are three parts to the installation:

- Oracle Installation
- Simulator Installation
- script installation and PDB ingest

Materials Needed:

- The CD containing the Oracle software
- One or more versions of the Aura Project Data Base (PDB)
- The CD containing the MPS/Aura Release 3.0 software

C-3.1: Oracle Installation

Installation of the Oracle database product need only be performed if the simulator is being installed on a new PC or one that has had its hard drive replaced. If Oracle is already installed on the PC then skip to Paragraph C-3.2.

Two separate Oracle product groups are required for the MPS/Aura simulator – Oracle 8i Enterprise 8.1.5.0.0 and Oracle 8i Programmer 8.1.5.0.0. Both product groups are contained on a single CD. There are a few points to keep in mind:

- The installing account must have administrator privileges.
- The machine selected as the Oracle server must have a D: drive.
- The server must be re-booted after installing both product groups.

Installation Instructions for Oracle8i Enterprise 8.1.5.0.0

Log in using the Administrator account. The entire installation should take no more than 10 minutes.

Panel	Panel Title	User Action
1.	Oracle8i Enterprise Edition for Windows NT	1. Select **Install/Deinstall Products**
2.	Oracle Universal Installer (O. U. I.)	1. Select **Next**
3.	File Locations	1. Source… Accept the default 2. Destination – Name Enter **DEFAULT_HOME** 3. Destination – Path Enter **D:\orant** 4. Select **Next**

Panel	Panel Title	User Action
4.	Available Products	1. Select **Oracle8i Enterprise Edition 8.1.5.0.0** 2. Select **Next**
5.	Installation Types	1. Select **Typical Configuration** 2. Select **Next**
6.	Location for Oracle Documentation	1. Select **Hard Drive** 2. Select **Next**
7.	Database Identification	1. Global Database Name Enter **oracle.world** 2. Select **Next**
8.	Summary	Select **Install**
8a.	Install	(Wait)
8b.	Configuration Tools	(Wait)
8c.	Oracle Database Configuration Assistant Alert	Click on **OK**.
9.	End of Installation	Select **Next Install**

Installation Instructions for Oracle8i Programmer 8.1.5.0.0

Panels 1 and 2 will only be seen if Oracle Programmer is being installed separately. If this is a continuation of the Oracle installation, control will go immediately to panel 3.

Panel	Panel Title	User Action
1.	Oracle8i Enterprise Edition for Windows NT	1. Select **Install/Deinstall Products**
2.	Oracle Universal Installer (O. U. I.)	1. Select **Next**
3.	File Locations	1. Source... **Accept the default** 2. Destination - Name Enter **DEFAULT_HOME** 3. Destination - Path Enter **D:\orant** 4. Select **Next**
4.	Available Products	1. Select **Oracle Programmer 8.1.5.0.0** 2. Select **Next**
5.	Installation Types	1. Select **Custom Installation** 2. Select **Next**
6.	Available Product Components	1. Select **ProC/C++** which automatically selects other items from the list. 2. Select **Next**
7.	Summary of Products	1. Select **Install**
8.	Install	**Wait** (When installation completes, the next panel pops up automatically.)
9.	End of Installation	1. Select **Exit** and Select “Yes” in the Confirmation pop-up box that appears.
10.		Close the Oracle Installation program by clicking on the X in the upper right corner.

All Oracle software required to run the MPS/Aura simulator has been installed. You must now re-boot your workstation and log in as the desired simulator user. The Oracle services will not begin running until you do this.

C-3.2: Java Runtime Engine Installation

Installation of the Java Runtime Engine product need only be performed if the simulator is being installed on a new PC or one that has had its hard drive replaced. If Java is already installed on the PC then skip to Paragraph C-3.3.

1. Insert the CD containing the MPS/Aura Release 3.0 into the CD drive and navigate to it using either Windows Explorer or My Computer.

2. Double-click on the file named **jdk1_2_2-win.exe.** This will cause the Java Runtime Engine to be installed. Accept all defaults when responding to the installation prompts.

C-3.3: Installation of the Aura Server and Client software

The steps in this paragraph cause the MPS/Aura Client and Server software to be installed on the PC.

1. Insert the delivery media into the appropriate drive.

2. To install the Aura Client:
   a) On the desktop, click on the Start button, and then select Run from the resulting menu.
   b) When the Run window appears select the Browse… button.
   c) From the Browse Window, select the Removable drive that contains the installation CD.
   d) Click on the Client folder.
   e) From within the Client folder, double click on the **Setup.exe** filename.
   f) A window with the title "Run Window" will appear. Click on the Okay button to proceed to the next step.
   g) The screen will be filled with an Aura Client background and a smaller window with the title "Welcome to Aura Client 3.0" will appear. Click on the Next button to proceed to the next step.
   h) The next window will contain the licensing agreement. Click on Yes to accept the agreement and proceed.
   i) After all of the files are copied, a window with the title "Setup Complete" will appear. Click on the Finish button to end.
   j) An Aura Client icon will now be installed on the desktop.

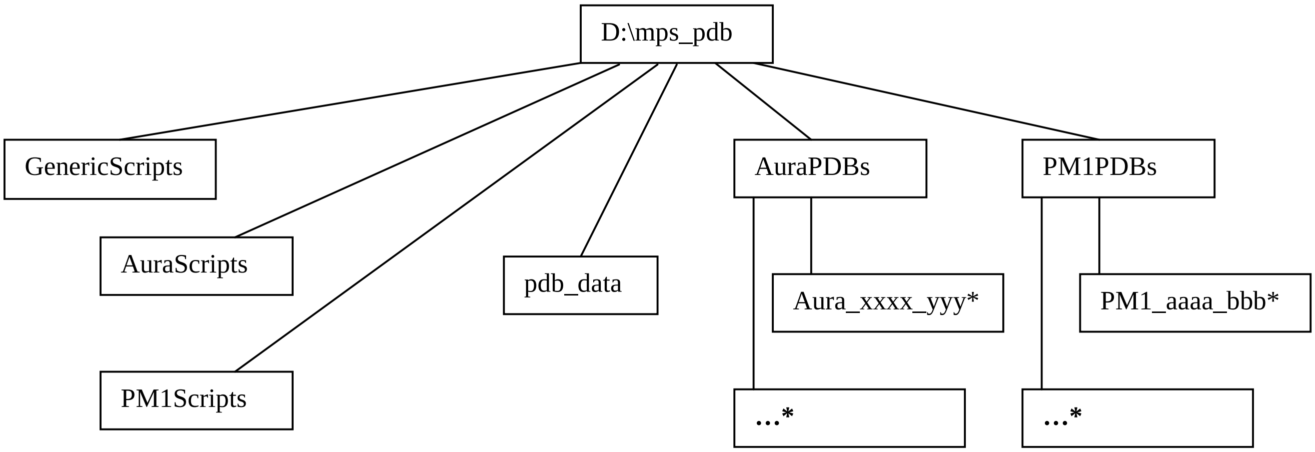
3. To install the Aura Server:
   a) On the desktop, click on the Start button, and then select Run from the resulting menu.
   b) When the Run window appears select the Browse… button.
   c) From the Browse Window, select the Removable drive that contains the installation CD.

d) Click on the Server folder.
e) From within the Server folder, double click on the **Setup.exe** filename.
f) A window with the title “Run Window” will appear. Click on the Okay button to proceed to the next step.
g) The screen will then be filled with an Aura Server background and a window with the title of “Welcome to Aura Server 3.0” will appear. Click the Next button to proceed.
h) The next window will contain the licensing agreement. Click on Yes to accept the agreement and proceed.
i) Next a window will show the completion status as the files are copied. When the copying is complete click on the Finish button to finish the installation.
j) An Aura Server icon will be installed on the desktop.

C-3.4: Installation of the Oracle scripts

Installation of the Oracle scripts need only be performed if the simulator is being installed on a new PC or one that has had its hard drive replaced. If the scripts are already installed on the PC then skip to Paragraph C-3.5. Re-installing the Oracle scripts will cause any PDB files previously placed on the hard drive to be deleted. However the contents of the Oracle database will not be affected unless the **CreateMPSPDB.bat** batch program or the **Build_PDB_Tables.bat** batch program is executed.

To ease the transition, a batch file is being provided which will create the new directory structure. When executed, it will create the following directory structure with needed batch files, SQL scripts, and Java programs, minus the PDB-specific folders, on the D drive of the hard disk. These directions create PDB tables within Oracle that will be useable by Release 6.5 of the MPS/Aqua simulator also.



* Folders containing Aqua or Aura PDB source files.

To install the database scripts:

a) Insert the delivery media into the appropriate drive.
b) On the desktop, click on the Start button, and then select Run from the resulting menu.
c) When the Run window appears select the Browse… button.
d) From the Browse Window, select the Removable drive that contains the installation CD.
e) Click on the Database folder.
f) From within the Database folder, double click on the **Database.bat** filename.

C-3.5: PDB Download

The next step is to copy the PDB onto the hard drive. You will need at least one version of the Aura PDB. The following PDB flat files are needed, where *xxxxxx* corresponds to the version portion of the filename:

cmd_desc_*xxxxxx*.pdb
cmd_fixdata_*xxxxxx*.pdb
cmd_parm_*xxxxxx*.pdb
cmd_vardata_*xxxxxx*.pdb
cmd_verify_*xxxxxx*.pdb
tlm_calcurve_*xxxxxx*.pdb
tlm_desc_*xxxxxx*.pdb
tlm_dstate_*xxxxxx*.pdb
tlm_packet_*xxxxxx*.pdb
tlm_parm_*xxxxxx*.pdb
tlm_polyconv_*xxxxxx*.pdb
tlm_rylim_*xxxxxx*.pdb

Add a folder to the directory structure shown above, under the **AuraPDBs** folder, to hold the source files of the Aura PDB.

Copy the desired version of the PDB into the folder just created. If desired, more than one version of the PDB may be copied. Be sure to copy each version into its own folder.

C-3.6: Database setup and PDB Ingest

The next steps are concerned with creation of the Oracle database and ingest of the PDB. If the action described has already been performed you may skip to the following step.

1. Create the simulator user account with userid **stest**, password **etsmps1:**

   Using either Windows Explorer or My Computer, navigate to the **D:\mps_pdb\GenericScripts** folder and double-click on the **CreateMPSPDB.bat** batch program.

2. Create the PDB tables within the Oracle database:

   Locate and double-click on the **Build_PDB_Tables.bat** batch file within the **D:\mps_pdb\GenericScripts** folder

3. Load the Aura PDB into Oracle:

   Repeat this step as many times as necessary to ingest all versions of the Aura PDB that you have placed on the hard drive.

   Note that the last PDB version ingested into Oracle will become the default.

   Using Windows Explorer (or My Computer), navigate to the **D:\mps_pdb\AuraScripts** folder, locate the **Load_Next_Aura_PDB.bat** batch file, and execute it by double-clicking on the filename. This batch file will ingest Aura PDBs into Oracle. It does not matter whether this is the first or any subsequent version of the PDB

   As the batch file executes it will call various batch programs, executable programs, Java programs, and SQL scripts in the **GenericScripts** and **AuraScripts** folders.

   1. During the batch file execution you will be prompted to specify the path to the PDB source files. Follow these steps to do so:

   - In the pop-up window titled **Please select a directory** click on the Browse button. That is the button with three dots on the right of the window.
   - In the resulting **File Open** window navigate through the directory structure by double-clicking on folder names until the desired folder appears. Do not open that folder. Select it by single-clicking on its name.
   - Next, single-click on the Open button. The **File Open** window will disappear.

- It may take a minute for the directory path to appear in the filename box of the **Please select a directory** window. When it does, click the **Close** button. All PDB files have now been copied to the **pdb_data** folder.

2. Also during the batch file execution you will be required to type in a version designator. You may enter 1 – 32 characters to uniquely identify this version of the PDB. All alphabetic characters will be translated to capital letters.

4. Ingest Verification

You may verify the correctness of the PDB ingest by checking the *.log files in the **D:\mps_pdb\AuraScripts** folder.

You may also check the ingest by using SQL*Plus to check the integrity of the PDB within Oracle. While there are many checks that may be run, the following two SQL queries should suffice.

Invoke SQL*Plus and enter the query:

select * from versions;

The version designator you entered should appear in the resulting list. Any alphabetic characters will have been translated to capital letters.

Enter the query:

select count(*) from cmd_full_fixed where version=’<your version designator>**’ and sc_name=’AURA’;**

The result should be a non-zero record count.

Attachment D - Special Operating Instructions

This attachment contains new special operating instructions for MPS/Aura Release 3.0. The information presented in this attachment has been checked for accuracy by the independent test team.

A User's Guide is being updated to include the information presented in this section. The User's Guide will be available from the ETS home page at http://esdis-it.gsfc.nasa.gov/ ETS/ets.html.

PDB Ingest Enhancements

The PDB schema described in the Aura PDB Data Format Control Document (DFCD) has been adopted for storing the database flat files into Oracle.

> MPS/Aqua Release 6.4 or earlier and MPS/Aura Release 2.0 will not be able to interface with Oracle once the new schema has been applied. It is recommended that only one PC be updated to the new schema. Directions are given below for accessing a PDB across a network.

The directions for creating the new PDB tables within Oracle are given in Attachment C, Paragraph C-3.6, steps 1 and 2. These directions create PDB tables that will be useable by Release 6.5 of the MPS/Aqua simulator also.

The following steps are to be followed to ingest each new PDB into Oracle once the tables have been defined.

Note that the last PDB version ingested into Oracle will become the default.

1. Create a folder under **D:\mps_pdb\AuraPDBs** and load the new source PDB files into it. Make no changes to the file names. The Java program that copies the files to the **pdb_data** folder expects the filenames to be in the format described in Paragraph 4.2.2.6 of the Aura DFCD. (Example: tlm_packet_xxxxxx.pdb, where xxxxxx is the version designator.)

2. Using Windows Explorer (or My Computer), navigate to the **D:\mps_pdb\AuraScripts** folder, locate the **Load_Next_Aura_PDB.bat** batch file, and execute it by double-clicking on the filename. This batch file will ingest Aura PDBs into Oracle. It does not matter whether this is the first or any subsequent version of the PDB

   As the batch file executes it will call various batch programs, executable programs, Java programs, and SQL scripts in the **GenericScripts** and **AuraScripts** folders.

3. During the batch file execution you will be prompted to specify the path to the PDB source files.
   - In the pop-up window titled **Please select a directory** click on the Browse button. That is the button with three dots on the right of the window.
   - In the resulting **File Open** window navigate through the directory structure by double-clicking on folder names until the desired folder appears. Do not open that folder. Select it by single-clicking on its name.
   - Next, single-click on the Open button. The **File Open** window will disappear.
   - After the directory path appears in the filename box of the **Please select a directory** window, click the **Close** button.
   - At this point the PDB files will be copied to the **pdb_data** folder, renamed, and modified to be ingested into Oracle.

4. Also during the batch file execution you will be required to type in a version designator. You may enter 1 – 32 characters to uniquely identify this version of the PDB. All alphabetic characters you enter in the version designator will be translated to capital letters.

Operation of the MPS/Aqua Simulator

MPS/Aqua Release 6.4 and earlier, and MPS/Aura Release 2.0, will not be able to interface with Oracle once the table schema is updated to the Aura format. If another PC with Oracle installed is connected to the same subnet, and has the Aqua version of the PDB schema installed, it is possible to run the Aqua simulator by connecting and loading the Aqua PDB across the network.

It is also possible to run the MPS/Aura Release 3.0 simulator on a PC that does not have the new Oracle table schema installed if it can be connected to Oracle across a network.

Follow these steps to set up the proper Oracle services.

To use this capability, Oracle must be installed on both computers and an alias for the remote server must be created on the PC where the simulator will run. This alias is called a Service Name. To create a Service Name, do the following:

1. Bring up **SQL*Plus** on the PC containing the Oracle database you wish to use and type “SELECT NAME FROM V$DATABASE;”. The last line of the resulting output will be the database name. You will need this name when performing step 7 below.

2. Determine the IP address of the PC containing the Oracle database.

   All remaining steps are performed on the PC where the simulator will run.

3. Start the Net8 Assistant by selecting **Start** -> **Programs** -> **Oracle** -> **Network Admin** -> **Net8 Assistant**. (The exact path may be different.)

4. Click the **Net Service Names** folder then select the *green* "+" in the upper left-hand corner to add a service.

5. The first panel prompts for a **Service Name**. It is just an alias so enter something short and meaningful then select Next.

6. Make sure **TCP/IP** (Internet Protocol) is highlighted and then select Next.

7. Enter the IP address of the remote database server in **Host Name** field, accept the default Port Number (1521), then select Next.

8. Select the *Oracle 8I release 8.0 or previous* radio button. Enter the name of the default database on the remote server in the **Oracle SID** field, then select Next.

9. Do **NOT** select **Test**. Click on **Finish**. You should see your new **Service Name** (with *.world* appended to it) under the **Net Service Names** folder.

10. Select **File** -> **Save Network Configuration**.

11. Select **File -> Exit.**

12. To test the connection, bring up **SQL*Plus** and attempt to connect to the remote server by entering your new **Service Name** in the **Host String** field (don't enter the *.world*). Enter the usual **User Name** and **Password**.

13. When initializing the simulator, enter the **Service Name** into the **Host** field of the **Load Database** window.

Local Variable Usage

The 26 local variables A through Z described in earlier releases have been updated to be true container items. This means that they may be viewed as well as set, and may be tested within scenario scripts. Additionally 26 new local variables, named AQ through ZQ, have been added to hold 64 bit integers.

These local variables are available only within the SCAura module.

An example of using these local variables within a scenario script follows:

```
; Conditional Scenario example using local variables
;
set CDH_SS_ISASUUSYNC 0

set IQ 0

sleep 1000
while (IQ < 10)

   set CDH_SS_ISASUUSYNC += 1
   sleep 1000

   if (MOD_CR_SR_GRAT_CH_B == 0)
         SET MOD_CR_SR_GRAT_CH_B 1
         set IQ += 1
   else
         set MOD_CR_SR_GRAT_CH_B 0
         set IQ += 2

         ; What follows is an example of saving a tlm value
         ; in case it changes
         set B = GNC_SS_STAOUT1W22
   endif
endwhile
; end
```

Telemetry Data Value Validation

The values of 1’s and 2’s complement signed integers, when entered by the operator, are now checked to ensure that the signed value will fit into the packet space. For example, given that a telemetry parameter is an 8-bit 2’s complement signed integer, any entry that is more negative than –128 or more positive than 127 will be rejected by the simulator with an error message. The parameter will be set to zero. A side effect of this capability is that hexadecimal numbers will always be taken as positive numbers.

Save and Restore of Display/Set Container Items

A set of mnemonics entered into a Display/Set Container Items window may now be saved for later use.

To save a set of container item names, click the **Save** button at the bottom of the window. You will be prompted to choose a path and enter a filename under which the set of mnemonics will be saved.

To restore a set of container item names, click the **Restore** button at the bottom of the window. You will be prompted to choose a path and filename from which the display will be restored. After restoring the display, click **Apply** to cause updates to begin.

Segmented Telemetry Transmission

The capability has been added for MPS/Aura to properly transmit TES segmented APIDs. The capability has been generalized so that any APID may be declared as being segmented in the future without any software change necessary. The following conditions must be met for MPS/Aura to transmit segmented telemetry packets:

- Only APIDs defined in the PDB can be transmitted.
- In addition, all segmented APIDs must be listed in **SegmentedApids.txt**, a new file which is stored in the **Aura Server 3.0\Properties** folder.

The entries in **SegmentedApids.txt** cause 5000 byte buffers to be created at initialization for each APID defined. A boilerplate file with TES APIDs 1715, 1716, and 1717 defined is being delivered with Release 3.0. It may be modified as necessary to include the APID numbers of all segmented telemetry packets. APIDs which are not segmented should not be listed in **SegmentedApids.txt**.

It is understood that none of the segmented APIDs will have telemetry points defined in the PDB. Therefore data needs to be set into the APID buffer prior to transmission. A boilerplate scenario script, **TES Boilerplate 3 segments.txt**, is being delivered with the release. It is set up to put data into APID 1715, stop the transmission of APID 1690, enable the transmission of APID 1715, wait for transmission of three segments, and finally disable APID 1715 and re-enable APID 1690. The setbuffer directive is used to fill the buffer with data. The scenario script contains complete directions for its usage. Directions for using the setbuffer directive may be found in Paragraph SIMSS-5.2.4.3 of the MPS/Aura User’s Guide.

Miscellaneous

- In response to user requests to make the initial setup easier, boilerplate Projects for Serial mode transmission and playback file creation are being delivered with the software.

- The PDB Red/Yellow Limits file is now used when setting initial telemetry values. If the telemetry parameter has an entry in the Red/Yellow Limits file, the simulator sets its value to the midpoint between the Yellow Low and Yellow High limits.

- The VCDU Sequence Counter field within APID 1000 uses a 32 bit space. The upper 8 bits are maintained at zero while the lower 24 bits track the VCDU Sequence Counter value.

- The Event Message window has been separated from the module window and made resizable. In addition, the Vertical Scroll Bar Slider will now stay where placed when it is moved away from the bottom of the Scroll Pane. To view new Event Messages, move the Scroll Bar Slider to the bottom of the Scroll Pane.

Attachment E – Resolved Discrepancy Reports

The following Discrepancy Reports (DRs) and Change Requests (CRs) have been closed by and are being delivered with MPS/Aura Release 3.0. In addition, an improvement is being made to the logic to answer DR ETS0363, which was closed in MPS/Aura Release 2.0. Although all of the DRs were written against the MPS/PM-1 simulator they apply to the MPS/Aura simulator as well. The DRs/CRs are listed in the table below, which provides the DR/CR Number, Status, Severity, and a short description. A full description of each DR/CR follows the summary table. Complete information on all DRs/CRs may be accessed via the Internet at address **http://edosultra30.gsfc.nasa.gov/ddts/**

Summary of Closed Discrepancy Reports

Critical (Severity 1)	Urgent (Severity 2)	Routine (Severity 3)	Change Requests	Total
0	0	3	0	3

Status Definitions

N – New	A – Assigned Analysis	R – Analysis Entered
V – Assigned Verification	T – Tested	C – Closed
W – Withdrawn	P – Postponed	X – Duplicate

ETS No.	SMO No.	Type	Severity	Description
ETS0438	SMOdr10435	DR	3	MPS initial values cause out-of-limit alarms at EMOS
ETS0443	SMOdr11451	DR	3	While loop iterations are limited
ETS0444	SMOdr11454	DR	3	Database load program cannot take spaces in pathname

```
DR:   ETS0438                  Related NCR:                Submitted: 010430
Status: ASSIGNED-ANALYSIS      Class: ETS               Asgnd-Analysis: 010611

  Title: MPS initial values cause out-of-limit alarms at EMOS
SUBMITTAL INFORMATION                    ANALYSIS INFORMATION
 Project:      ETS                        Assignee1/Org: Ernest Quintin
 DR Type:    Problem                      Phone: 301-805-3649
 Rel/Ver:      6.2                        Email: equintin@csc.com
 Subsystem:    MPS-PM/Aqua                Assignee2/Org:
 Module:    Simulator                     Phone:
 Affected-Requirement:                    Email:
 Test Phase:   in-field use               Date due (Sev=1,2):
 Severity:     3
 Date found:   010427
 Location:     Denver
 Submitter:    Ernest Quintin
 Organization: ETS Dev Group
 Phone number: 301-805-3649
 Email:    equintin@csc.com

************ Problem (Added 010430 by equintin) ***********
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

MPS presets all telemetry points to a midpoint of possible data
values.  This causes numerous out-of-limit alarms at EMOS when
MPS telemetry data is initialized.

Reported by Vince Ruland.
```

```
DR:   ETS0443                 Related NCR:                    Submitted: 010719
Status: NEW                   Class: ETS

  Title: While loop iterations are limited
SUBMITTAL INFORMATION
 Project:      ETS
 DR Type:    Problem
 Rel/Ver:      6.3
 Subsystem:    MPS-PM/Aqua
 Module:    Simulator
 Affected-Requirement:
 Test Phase:   ops test
 Severity:     3
 Date found:   010629
 Location:     GSFC
 Submitter:    Ernest Quintin
 Organization: ETS Dev Group
 Phone number: 301-805-3649
 Email:    equintin@csc.com
```

************ Problem (Added 010719 by equintin) ***********

Please describe the problem you are experiencing below, including what you did, what you expected to happen, and what actually happened:

Testing of scenario scripts requested by the FOT for SSR emulation has shown that While loops cannot run for more than about 1000 iterations without crashing the scenario dll executable.

A workaround is to exit the loop, then call the scenario script from itself.

```
DR:   ETS0444                 Related NCR:                    Submitted: 010719
Status: NEW                   Class: ETS

  Title: Database load program cannot take spaces in pathname
SUBMITTAL INFORMATION
 Project:      ETS
 DR Type:    Problem
 Rel/Ver:      6.3
 Subsystem:    MPS-PM/Aqua
 Module:    Simulator
 Affected-Requirement:
 Test Phase:   acceptance test
 Severity:     3
 Date found:   010629
 Location:     GSFC
 Submitter:    Ernest Quintin
 Organization: ETS Dev Group
 Phone number: 301-805-3649
 Email:    equintin@csc.com

************ Problem (Added 010719 by equintin) ***********
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

The Java program that copies PDB files and truncates their names for
ingest into Oracle cannot locate the source folder if there are spaces
in the pathname.
```

The following DR was corrected in MPS/Aura Release 2.0. Additional logic has been added to the code to validate one's and two's complement signed integers.

```
DR:   ETS0363                 Related NCR:                    Submitted: 991217
Status: ASSIGNED-ANALYSIS     Class: ETS                 Asgnd-Analysis: 000117

  Title: Telemetry data values entered by operator are not validated
SUBMITTAL INFORMATION                     ANALYSIS INFORMATION
 Project:      ETS                         Assignee1/Org: Ernest Quintin
 DR Type:    Problem                       Phone: 301-805-3649
 Rel/Ver:      3.0                         Email: equintin@csc.com
 Subsystem:    MPS-PM/Aqua                 Assignee2/Org:
 Module:    Simulator                      Phone:
 Affected-Requirement:                     Email:
 Test Phase:   acceptance test             Date due (Sev=1,2):
 Severity:     3
 Date found:   991217
 Location:     GSFC
 Submitter:    Ernest Quintin
 Organization: ETS Dev Group
 Phone number: 301-805-3649
 Email:    equintin@csc.com

************ Problem (Added 991217 by equintin) ***********
No validation is being performed on the values entered in the
Set Directive Window. If the value entered exceeds the number
of bits specified for the telemetry point, high order bits are
truncated when the packet is built. If you enter a hex value,
the telemetry point gets set to zero. Binary values are
interpreted as decimal. In addition, the system does not notify
the user concerning the invalid entry



************ Admin Comment (Modified 010402 by ckolb) ***********
3/30/2001 - Per DRB, this problem will be fixed in Aura Release 2.

01/05/01 DRB: Per developer, the fix for this will be included
in Release 6.2, scheduled for January 26 delivery.
```

Attachment F – Unresolved Discrepancy Reports

All of the Discrepancy Reports (DRs) and Change Requests (CRs) listed in the following table were written against the MPS/Aqua simulator. They are being carried forward because the software has been ported to MPS/Aura from that simulator. The table includes the DR/CR Number, Status, Severity, and a short description. A full description of each DR/CR follows the summary table. Complete information on all DRs/CRs may be accessed via the Internet at **http://edosultra30.gsfc.nasa.gov/ddts/**

Summary of Open Discrepancy Reports

Critical (Severity 1)	Urgent (Severity 2)	Routine (Severity 3)	Change Request (CR)	Total
0	0	2	0	2

Status Definitions

N – New
V - Assigned Verification
W – Withdrawn

A - Assigned Analysis
T – Tested
P – Postponed

R - Analysis Entered
C – Closed
X – Duplicate

ETS #	SMO No.	Type	Severity	Status	Description
ETS0392	SMOdr06633	DR	3	A	Loading database
ETS0423	SMOdr08499	DR	3	A	MPS-1 Crash During MODIS IOE-1

```
DR:   SMOdr06633 (ETS0392)    Related NCR:      Submitted: 000404
Status: ASSIGNED-ANALYSIS     Class: ETS        Asgnd-Analysis: 000512

  Title: Loading database
SUBMITTAL INFORMATION                     ANALYSIS INFORMATION
 Project:         ETS                      Assignee1/Org: Ernest Quintin
 Rel/Ver:         4.0                      Phone: 301-805-3649
 Subsystem:       MPS-PM/Aqua              Email: equintin@csc.com
 Test Phase:      unit test                Assignee2/Org:
 Severity:        3                        Phone:
 Date found:      000331                   Email:
 Location:        Denver                   Date due (Sev=1,2):
 Submitter:       Vince Ruland
 Organization:    EMOS
 Phone number:    720-895-4068
 Email:           vhruland@west.raytheon.com

************ Problem (Added 000404 by vruland) ***********
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:


A user should be able to load a new version of the database into
the SC module without having to cycle the entire project first.

Whenever the apply button is clicked for the Load Database on a
SC module, the previous instance of the database should be purged
before the module loads the new instance.


************ Admin Comment (Modified 000515 by eshurie) ***********
At 5/12/00 DRB, the developer got more information on this
request, and will write a memo assessing the difficulty of the
different programming options.

Per DRB on 4/14/00, this DR is on HOLD pending further
contemplation of it by Denver.  (Would not be able to be
done within a week.)
```

```
DR:   SMOdr08499 (ETS0423)    Related NCR:            Submitted: 001113
Status: ASSIGNED-ANALYSIS     Class: ETS              Asgnd-Analysis: 001208

  Title: MPS-1 Crash During MODIS IOE-1
SUBMITTAL INFORMATION                     ANALYSIS INFORMATION
 Project:         ETS                      Assignee1/Org: Ernest Quintin
 DR Type:         Problem                  Phone: 301-805-3649
 Rel/Ver:         6.0                      Email: equintin@csc.com
 Subsystem:       MPS-PM/Aqua              Assignee2/Org:
 Module:          Simulator                Phone:
 Affected-Requirement:                     Email:
 Test Phase:      unit test                Date due (Sev=1,2):
 Severity:        3
 Date found:      001107
 Location:        GSFC
 Submitter:       MODIS IOT
 Organization:    Other
 Phone number:    301-614-5025
 Email:           modiot@mcst.gsfc.nasa.gov
```

************ Configuration (Added 001113 by MODIOT) ***********
Please describe the current system configuration:

The MPS-1 (Located in the ISR) was brought up on port A4 to be used by the SIM-S configuration. The predefined project for server/client version 6.0 was loaded and database version 000822-005 was ingested. The simulator was configured for nominal 16kbps telemetry. Several scenario files were run via the Command-Scenario capability of the simulator to mimic the functionality of MODIS.

************ Problem (Added 001113 by MODIOT) ***********
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

The MPS-1 ran smoothly for the first five hours of the test. There were no indications of any problems with the simulator.

Abruptly, the server component of the software crashed, leaving the client component disconnected and "lost". The "Dr. Watson" monitoring software caught the crash and produced the standard Windows NT illegal exception violation report. At the time of the crash, the only activity the simulator was performing was standard telemetry broadcasting (no scenario files were executing, and no commands were being interpreted).

Attempts to bring up another instance of the server component for the client component to reconnect to failed to succeed. The client component would not reconnect. The client component had to be exited, and a new instance started to successfully connect to the new server component.

The result of recycling the client component was the complete loss of the state of the simulator at the time of the crash. The client component will reset all telemetry to a default value when it is started, thereby negating all activities that had been previously performed during the IOE. It took a significant amount of time to restore the simulator's state to what it was prior to the server component crash.

The cause for this software crash was not obvious, and is presently still a mystery. After restarting the server component, it operated without incident for the remainder of the MODIS IOE-1.

[In the past, it has been noted that the MPS server and client components appear to have a memory leak. When the software exits, it does not release the memory allocated to it, etc. Perhaps the crash was caused by over five hours of slow memory leaks. This would explain why the incident did not occur again: the test finished in less than 3 hours after the server component was restarted.]

Attachment G: Requirements Matrix

The following Level 4 Requirements list has been adapted from the Aura Level 4 Requirements list delivered with Release 2.0. Many of the capabilities resulting from these requirements were ported from the Aqua simulator rather than representing work done to produce the Aura simulator. Those requirements that were satisfied in Release 1.0 and are new to the Aura simulator are marked with an asterisk (*) in the Release column.

The Date of Insert/Update column may be used together with the Comments column to identify changed requirements and provide easy traceability. When a requirement is changed, a new entry, without a requirement number, is added just below the existing entry. Rather than being deleted, the original entry will be marked as being superceded. Original entries may be removed in later versions of the document.

Date of this update: 9/28/2001
General changes: Removed the comments "NEW", "REWORDED", and "RENUMBERED" from certain requirements.
Removed those requirements which had previously been marked "SUPERCEDED".

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
CMD-01	1	The MPS/Aura simulator shall be capable of receiving command data as UDP command blocks.		
CMD-02	1	The MPS/Aura simulator shall be capable of configuring command receipt processing in IP mode.		
CMD-02.01	1	The MPS/Aura simulator shall be capable of configuring IP mode command receipt to UDP MULTICAST mode.		
CMD-02.02	1	The MPS/Aura simulator shall be capable of configuring IP mode command receipt to any valid UDP MULTICAST IP address.		
CMD-02.03	1	The MPS/Aura simulator shall be capable of configuring IP mode command receipt to any valid UDP MULTICAST Port number.		
CMD-02.04	1	The MPS/Aura simulator shall be capable of configuring IP mode command receipt to any block length between one and 6000 bytes.		
CMD-03	1	The MPS/Aura simulator shall accept operator directives that enable or disable the following elements of the command validation process: Codeblock BCH Parity Validation, Transfer Frame Header Validation, FARM Protocol Validation, and User Command Packet Header Validation.		
CMD-03.01	1	When the Codeblock BCH parity validation element is enabled, the command subsystem will verify for each codeblock of each received CLTU that the BCH parity byte matches a computed value and that the spare bit is equal to zero. If any codeblock of a CLTU fails validation, an event message will be generated and that entire CLTU will be discarded. When this element is disabled, the parity byte will be assumed to be valid.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
CMD-03.02	1*	When the Transfer Frame Header validation element is enabled, the command subsystem will verify that all of the fields of the Transfer Frame header, except the sequence number, match expected values and ranges as defined in the ICD. If the Transfer Frame Header validation fails, an event message will be generated and the entire Transfer Frame will be discarded. If applicable, the CLCW corresponding to that Transfer Frame's virtual channel will be updated with error information. When this element is disabled, the Transfer Frame header values will be assumed to be valid.	3/7/2001	Spacecraft ID was changed to 0xCC; otherwise no change from Aqua.
CMD-03.03	1	When the FARM validation element is enabled, the command subsystem will verify that the Transfer Frame sequence number is valid as expected for FARM-1 protocol as defined in the ICD. If the FARM validation fails, an event message will be generated and the entire Transfer Frame will be discarded. If applicable, the CLCW corresponding to that Transfer Frame's virtual channel will be updated with error information. When this element is disabled, the Transfer Frame sequence number will be assumed to be valid.		
CMD-03.04	1	When the Command Packet Header validation element is enabled, the command subsystem will verify that the Command Packet Header fields contain valid values as defined in the ICD. If the Command Packet Header validation fails, an event message will be generated and the Command Packet will be discarded. This requirement is applicable to the spacecraft command packet format and the instrument command packet format. When this element is disabled, the Command Packet Header is assumed to be valid.		
CMD-04	1	The MPS/Aura simulator shall accept operator directives to change all fields of the spacecraft and instrument CLCWs.		
CMD-05	1	The MPS/Aura simulator shall simulate spacecraft command acceptance according to the COP-1 protocol.		
CMD-05.01	1	The MPS/Aura simulator shall perform Type AD spacecraft command acceptance checks in accordance with the FARM-1 protocol if FARM-1 protocol checking is enabled.		
CMD-05.01.1	1	The MPS/Aura simulator shall reject Type AD spacecraft commands and post a command rejected event message if the Lockout bit is set in the spacecraft CLCW.		
CMD-05.01.2	1	The MPS/Aura simulator shall reject Type AD spacecraft commands, post a command rejected message, and set the Lockout bit in the spacecraft CLCW if (1) the Frame Sequence Count in the Transfer Frame header is more than 90 counts greater than or more than 90 counts less than (modulo 256) the Report Value field of the spacecraft CLCW and (2) FARM-1 protocol checking is enabled.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
CMD-05.01.3	1	The MPS/Aura simulator shall reject Type AD spacecraft commands, post a command rejected message, and set the Retransmit bit in the spacecraft CLCW if (1) the Frame Sequence Count in the Transfer Frame header is between one and 90 counts greater than (modulo 256) the contents of the Report Value field of the spacecraft CLCW and (2) FARM-1 protocol checking is enabled.		
CMD-05.01.4	1	The MPS/Aura simulator shall reject Type AD spacecraft commands and post a command rejected message if (1) the Frame Sequence Count in the Transfer Frame header is between one and 90 counts less than (modulo 256) the contents of the Report Value field of the spacecraft CLCW and (2) FARM-1 protocol checking is enabled.		
CMD-05.01.5	1	The MPS/Aura simulator shall clear the spacecraft CLCW Lockout bit upon receipt of an UNLOCK Control Command (Type BC) containing the spacecraft VCID.		
CMD-05.01.6	1	The MPS/Aura simulator shall set the spacecraft CLCW Report Value field to the data value contained within the third byte of a SET V(R) Control Command (Type BC) containing the spacecraft VCID.		
CMD-05.01.7	1	The MPS/Aura simulator shall increment the Report Value field (modulo 256) of the spacecraft CLCW upon receipt of a Type AD spacecraft command whose Frame Sequence Count matches the current spacecraft CLCW Report Value field contents, provided that FARM-1 protocol checking is enabled.		
CMD-05.02	1	The MPS/Aura simulator shall perform Type AD instrument command acceptance checks in accordance with the FARM-1 protocol if FARM-1 protocol checking is enabled.		
CMD-05.02.1	1	The MPS/Aura simulator shall reject Type AD instrument commands and post a command rejected event message if the Lockout bit is set in the instrument CLCW.		
CMD-05.02.2	1	The MPS/Aura simulator shall reject Type AD instrument commands, post a command rejected message, and set the Lockout bit in the instrument CLCW if (1) the Frame Sequence Count in the Transfer Frame header is more than 90 counts greater than or more than 90 counts less than (modulo 256) the Report Value field of the instrument CLCW and (2) FARM-1 protocol checking is enabled.		
CMD-05.02.3	1	The MPS/Aura simulator shall reject Type AD instrument commands, post a command rejected message, and set the Retransmit bit in the instrument CLCW, if (1) the Frame Sequence Count in the Transfer Frame header is between one and 90 counts greater than (modulo 256) the Report Value field of the instrument CLCW and (2) FARM-1 protocol checking is enabled.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
CMD-05.02.4	1	The MPS/Aura simulator shall reject Type AD instrument commands and post a command rejected message if (1) the Frame Sequence Count in the Transfer Frame header is between one and 90 counts less than (modulo 256) the Report Value field of the instrument CLCW and (2) FARM-1 protocol checking is enabled.		
CMD-05.02.5	1	The MPS/Aura simulator shall clear the instrument CLCW Lockout bit upon receipt of an UNLOCK Control Command (Type BC) containing the instrument VCID.		
CMD-05.02.6	1	The MPS/Aura simulator shall set the instrument CLCW Report Value field to the data value contained within the third byte of a SET V(R) Control Command (Type BC) containing the instrument VCID.		
CMD-05.02.7	1	The MPS/Aura simulator shall increment the Report Value field (modulo 256) of the instrument CLCW upon receipt of a Type AD instrument command whose Frame Sequence Count matches the current instrument CLCW Report Value field contents, providing that FARM-1 protocol checking is enabled.		
CMD-06	1	The MPS/Aura simulator shall provide the capability to monitor and display command processing status.		
CMD-07	1	Upon operator request, the MPS/Aura simulator shall store received commands for posttest review subject to specified storage capacities.		
CMD-08	1	The MPS/Aura simulator shall use information from the PDB to perform command identification processing. The Command subsystem shall match command bit patterns received to stored bit patterns to locate command mnemonics in the PDB.		
CMD-09	1	The MPS/Aura simulator shall provide the capability to respond to that subset of spacecraft commands that are defined in the Aura PDB Command Execution Verification (CEV) file. If the PDB CEV file contains end-item verifier telemetry mnemonics associated with the identified command, the associated telemetry point(s) will be set to the corresponding value(s) defined in the CEV file.		
CMD-10	1	The MPS/Aura simulator shall generate a simulator event message whenever a command is received.		
CMD-10.01	1	The MPS/Aura simulator shall generate a simulator event message to display the command mnemonic whenever a valid command is decoded.		
CMD-10.02	1	The MPS/Aura simulator shall generate a simulator event message to display the values of command submnemonics whenever a command containing submnemonics is decoded.		
CMD-10.03	1*	The MPS/Aura simulator shall provide for the storage of command submnemonic values, to be viewable by the operator, for the life of a simulations session.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
CMD-11	1	The MPS/Aura simulator shall generate a simulator event message whenever a command error is detected		
CMD-11.01	1	The MPS/Aura simulator shall generate a simulator event message indicating the command error detected whenever a command in error is decoded, provided that command validation is enabled.		
CMD-11.02	1	The MPS/Aura simulator shall generate an event message indicating that an unknown command has been received whenever a command cannot be matched to any PDB entry.		
CMD-12	1	The MPS/Aura simulator shall perform verification of selected fields of the Command Data Block (CDB) header of received commands. The fields to be verified shall be Message Type, Source, Destination, spacecraft identifier (SCID), and Sequence Count.		
CMD-12.01	1	The MPS/Aura simulator shall generate event messages reporting inconsistencies in the verifiable fields of the CDB.		
CMD-12.02	1*	The MPS/Aura simulator shall accept and execute operator directives that set expected values for verification of the CDB header.	3/7/2001	Added logic to EOSGS to accept expected Spacecraft ID
CMD-12.03	1	The MPS/Aura simulator shall permit the operator to enable and disable CDB verification.		
CMD-13	1	The MPS/Aura simulator shall receive spacecraft memory and table loads via command blocks and shall store the load data in a load buffer (simulated spacecraft memory.)		
CMD-13.01	1	The MPS/Aura simulator shall perform a validation of the command load data checksum, for those loads that contain a checksum.		
CMD-13.02	1	The MPS/Aura simulator shall permit the operator to inhibit the checksum validation.		
CMD-14	1	The MPS/Aura simulator shall be capable of simulating a spacecraft memory dump of loaded data.		
CMD-14.01	1	The MPS/Aura simulator shall be capable of copying a single memory load from the load buffer to the dump buffer.		
CMD-14.02	1	The MPS/Aura simulator shall permit the operator to inhibit copying the memory load to the dump buffer.		
CMD-15	1	The MPS/Aura simulator shall process commands that request or configure for a spacecraft memory dump.		
CMD-16	1	The MPS/Aura simulator interface with EOC shall comply with the command interface formats and protocols specified in the EDOS to EGS Elements interface document		
CMD-17	1	The MPS/Aura simulator shall be capable of updating multiple command counters in telemetry. The command counters to be updated shall be as agreed upon with Aura project representatives.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
CMD-18	1	The MPS/Aura simulator shall interpret VCID 0 (spacecraft), VCID 1 (instrument), and VCID 16 and 17 (TIE critical) commands.		
CMD-19	1	The MPS/Aura simulator shall be capable of interpreting multipart commands.		
CMD-20	1	The MPS/Aura simulator shall be capable of logging up to 8 MB of received commands during a testing session.		
CMD-21	1	The MPS/Aura simulator shall be capable of receiving spacecraft commands in a CLTU bitstream through the serial interface at rates from 125 bps to 2 Kbps.		
CMD-22	3	The MPS/Aura simulator shall be capable of receiving spacecraft command packets via a 1553B bus interface.	9/28/2001	See also requirement GEN-21. Implementation delayed due to delay in ETSF availability.
CMD-22.01	3	The MPS/Aura simulator shall be capable of connecting to multiple Remote Terminal ports when configured to accept commands via a 1553B bus interface.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.
CMD-22.02	3	The MPS/Aura simulator shall be capable of maintaining a count of commands received from each active Remote Terminal when configured to accept commands via a 1553B bus interface.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.
CMD-22.03	3	The MPS/Aura simulator shall be capable of bypassing Codeblock BCH Parity Validation, Transfer Frame Header Validation, FARM Protocol Validation, and User Command Packet Header Validation when configured to accept commands via a 1553B bus interface.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.
CMD	TBD	The MPS/Aura simulator shall be capable of a <TBN> simulation of spacecraft Stored Command Processing	3/7/2001 9/28/2001	Proposed. TBN. To Be Removed. Capability is not required by project.
CMD	TBD	The MPS/Aura simulator shall be capable of receiving instrument memory and table loads via command blocks and shall store the load data in a load buffer (simulated memory.)	3/7/2001	Proposed. Exact capability is TBN.
GEN-01	1	The MPS/Aura simulator shall be Year 2000 compliant		
GEN-02	1	The MPS/Aura simulator shall be capable of maintaining an internally generated time code to a resolution of 125 milliseconds..		
GEN-02.01	1	The MPS/Aura simulator shall be capable of setting GMT and simulated spacecraft time as directed by the operator.		
GEN-03	1	The MPS/Aura simulator shall be capable of executing a scenario script file.		
GEN-03.01	1	The MPS/Aura simulator shall be capable of executing operator directives via a scenario script to update telemetry parameters by mnemonic.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
GEN-03.02	1	The MPS/Aura simulator shall be capable of executing operator directives via a scenario script to retrieve and display the value of any telemetry parameter by mnemonic.		
GEN-03.03	1	The MPS/Aura simulator shall be capable of executing operator directives via a scenario script to start and stop telemetry transmission.		
GEN-03.04	1	The MPS/Aura simulator shall be capable of executing operator directives via a scenario script to start and stop transmission of CLCW packets.		
GEN-03.05	1	The MPS/Aura simulator shall be capable of executing operator directives via a scenario script to enable and disable all elements of command validation that are under operator control. See "CMD" requirements for those command validation elements that are controllable by the operator.		
GEN-03.06	TBD	The MPS/Aura simulator shall provide a scenario file container name verification capability, the purpose of which is to verify that all telemetry and command mnemonics appearing in scenario scripts exist in the PDB.	5/17/2001	Proposed.
GEN-04	1	The MPS/Aura simulator shall be capable of providing files of received or generated test data on electronic and physical media.		
GEN-05	1	The MPS/Aura simulator shall acknowledge an operator request within 2 seconds of its entry.		
GEN-06	1	The MPS/Aura simulator shall start execution of an operator request within 5 seconds of its entry.		
GEN-08	1	The MPS/Aura simulator shall comply with the set of display guidelines specified in DSTL-92-007, Human-Computer Interface Guidelines, August, 1992.		
GEN-09	1	The MPS/Aura simulator shall comply with security provisions specified in the NASA Automated Information Security Handbook, NHB 2410.9A.		
GEN-10	1	The MPS/Aura simulator shall comply with the NASA Communications (Nascom) Access Protection Policy and Guidelines.		
GEN-11	1	The MPS/Aura simulator shall provide a hard disk drive with sufficient capacity to store the program bootstrap, executable files, and other simulation environment files, such as the Project Data Base and scenario files used during tests, and a TBD percent reserve.		
GEN-12	1	The MPS/Aura simulator shall provide a physical media storage device that can be used to support the exchange of small amounts of information with external systems and for system backups and data logging.		
GEN-13	1	The MPS/Aura simulator shall be portable.		
GEN-14	1	The MPS/Aura simulator shall provide an Ethernet interface that conforms to 10BaseT of the IEEE 802.3 standard.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
GEN-15	1	The MPS/Aura simulator shall interface with the EOC through the Ethernet interface using the Internet Protocol (IP) suite, including TCP/IP and UDP/IP.		
GEN-16	1	The MPS/Aura simulator shall receive CLTUs in command data blocks from the EOC and output EDUs (packets and CLCWs) to the EOC through the Ethernet interface. All data transfers through this Ethernet interface shall be based on UDP/IP protocol.		
GEN-17	1	The MPS/Aura simulator shall provide a capability to permit modification of the Command End-Item Verifiers file.		
GEN-17.01	1	The MPS/Aura simulator Command End-Item Verifiers modification utility shall permit the addition of a single record to the file of end-item verifiers at each invocation.		
GEN-17.02	1	The MPS/Aura simulator Command End-Item Verifiers modification utility shall permit the modification of a single record of the file of end-item verifiers at each invocation. The Low Limit and State Text fields shall be the only fields that may be modified.		
GEN-17.03	1	The MPS/Aura simulator Command End-Item Verifiers modification utility shall permit the deletion of a single record from the file of end-item verifiers at each invocation.		
GEN-18	1	The MPS/Aura simulator shall provide a file selection browse capability.		
GEN-19	1	The MPS/Aura simulator shall provide a configuration save and restore capability.		
GEN-19.01	1	The MPS/Aura simulator shall be capable of saving module configuration information. The information saved shall consist of the modules that constitute a project, and the module links and link source/destination numbers.		
GEN-19.02	1	The MPS/Aura simulator shall be capable of saving configuration information for IP modules, log modules, and Serial modules.		
GEN-19.03	1	The MPS/Aura simulator shall be capable of saving multiple configurations in separate disk files.		
GEN-19.04	1	The MPS/Aura simulator shall permit the operator to name a disk file in which configuration information shall be saved.		
GEN-19.05	1	The MPS/Aura simulator shall permit the operator to restore configuration information upon initialization.		
GEN-19.06	1	The MPS/Aura simulator shall be capable of restoring configuration information from an existing named disk file.		
GEN-19.07	1	The MPS/Aura simulator shall be capable of displaying the names of the disk files containing configuration information when responding to a restore request during initialization.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
GEN-19.08	TBD	The MPS/Aura simulator shall be capable of displaying the file creation date when responding to a restore request during initialization.		This is a function of the SIMSS baseline. The SIMSS release it will be in has not yet been decided.
GEN-20	1	The MPS/Aura simulator shall be capable of executing multiple scenario script files simultaneously, up to the limit imposed by CPU and memory capacities.		
GEN-20.01	1	The MPS/Aura simulator shall permit the operator full control of scenario script files that the operator has invoked. The control directives available shall consist of START, STOP, PAUSE, and RESUME.		
GEN-20.02	1	The MPS/Aura simulator shall be capable of starting a scenario script in response to a command received. The operator shall not be able to STOP, PAUSE, or RESUME a scenario script started this way.		
GEN-20.03	1	The MPS/Aura simulator shall be capable of invoking a scenario script from within a scenario script. The operator shall not be able to STOP, PAUSE, or RESUME a scenario script started this way.		
GEN-20.04	1	The MPS/Aura simulator shall be capable of displaying the status of all scenario scripts that were started by the operator. This status shall consist of an indication as to whether the scenario script is running, paused, or finished, a display of the current line number, and a display of the directive currently being executed.		
GEN-20.05	2	The MPS/Aura simulator shall be capable of executing a scenario script that contains nested if/then/else and while loop constructs.	5/17/2001	
GEN-20.06	2	The MPS/Aura simulator Scenario script processor shall be capable of connecting to up to four modules and sending script directives to the proper module. The operator must provide a module indicator.	5/17/2001	
GEN-20.06.1	2	The module indicator format in scenario scripts shall be of the form #n, where n is the link number.	5/17/2001	
GEN-20.06.2	2	When starting execution of any scenario script, the MPS/Aura simulator Scenario script processor shall default the module indicator to link #1. In the absence of a module indicator in a scenario script, the script processor shall send all directives to the module connected to link #1.	5/17/2001	
GEN-20.06.3	2	When a module indicator is encountered, the MPS/Aura simulator Scenario script processor shall send that and all following directives to the link number indicated, until another module indicator is encountered.	5/17/2001	

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
GEN-21	3	The MPS/Aura simulator shall provide a 1553B bus interface for the purpose of receiving command packets and transmitting telemetry packets.	9/28/2001	See GUI-17, CMD-22, and TLM-37 requirements for further details. Implementation delayed due to delay in ETSF availability.
GEN-21.01	3	The MPS/Aura simulator shall be capable of emulating up to four Remote Terminals when connected to a 1553B bus.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.
GEN-21.02	3	The MPS/Aura simulator, when connected to a 1553B bus, shall be capable of maintaining and displaying status counts of the number of commands received and telemetry packets transmitted through each Remote Terminal.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.
GEN	TBD	The MPS/Aura simulator shall be capable of selecting packets or VCDUs to be logged from a stream of mixed packets or VCDUs based on <TBN> criteria.	9/28/2001	Proposed. SUPERCEDED by GEN-22.
GEN-22	3	The MPS/Aura simulator shall be capable of selecting VCDUs to be logged from a stream of mixed VCDUs based on Virtual Channel.	9/28/2001	Implemented via SIMSS vcProcessor module.
GEN	TBD	The MPS/Aura simulator shall provide a limited command generation capability, the purpose of which is to facilitate verification of simulator operational readiness.		
GEN	TBD	The MPS/Aura simulator shall provide a limited telemetry data quality monitoring capability, the purpose of which is to facilitate verification of simulator operational readiness.		
GUI-01	1	The MPS/Aura simulator shall accept and validate all operator directives.		
GUI-01.01	1	The MPS/Aura simulator GUI shall maintain a history list of directives entered by the operator. This history list shall store a maximum of 10 operator directives.		
GUI-01.02	1	The MPS/Aura simulator GUI shall permit the operator to re-execute directives stored in the history list.		
GUI-01.03	1	The MPS/Aura simulator GUI shall permit the operator to edit directives stored in the history list.		
GUI-02	1	The MPS/Aura simulator GUI shall provide the capability to display command packets received.		
GUI-03	1	The MPS/Aura simulator GUI shall provide the capability to display telemetry and CLCW packets transmitted.		
GUI-04	1	The MPS-Aura simulator GUI shall provide the capability to display command and telemetry status.		
GUI-05	1	The MPS/Aura simulator GUI shall provide the capability to display the current receive and transmit network configuration to the operator.		

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
GUI-07	1	The MPS/Aura simulator shall provide the capability to display the EDOS Service Header appended to transmitted telemetry packets.		
GUI-08	1	The MPS/Aura simulator shall provide the capability to display the Telemetry Packet Header of a selected APID.		
GUI-09	1	The MPS/Aura simulator shall provide the capability to display GMT and Spacecraft times.		
GUI-10	1	The MPS/Aura simulator shall provide the capability to display the current values of the spacecraft and instrument CLCWs.		
GUI-11	1	The MPS/Aura simulator shall provide the capability to display event messages.		
GUI-11.01	1*	The MPS/Aura simulator shall provide the operator with the capability to suppress display of event messages. The filtering mechanism shall be keyed to the event message color.		
GUI-11.02	1	The MPS/Aura simulator shall log all generated event messages to a disk file.		
GUI-12	1	The MPS/Aura simulator shall provide the capability to display telemetry and CLCW transmit status.		
GUI-13	1	The MPS/Aura simulator shall provide the capability to display command receipt status.		
GUI-14	1	The MPS/Aura simulator shall be capable of updating all displays periodically.		
GUI-15	1	The MPS/Aura simulator shall provide a generic buffer display.		
GUI-16	2	The MPS/Aura simulator shall provide a single display giving the telemetry packet enable status and transmit interval for all APIDs in the PDB.		
GUI-17	3	The MPS/Aura simulator shall provide configuration and status displays for the 1553B bus interface.	5/17/2001 9/28/2001	NEW. See CMD-22, GEN-21, and TLM-37 requirements for further details. Implementation delayed due to delay in ETSF availability.
GUI-17.01	3	The MPS/Aura simulator 1553B bus interface configuration display shall permit the operator to enter one to four Remote Terminal numbers and associate a range of telemetry APIDs to be transmitted over each.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.
GUI-17.02	3	The MPS/Aura simulator 1553B bus interface status display shall display the number of command packets received and telemetry packets transmitted for each Remote Terminal connected.	5/17/2001 9/28/2001	Implementation delayed due to delay in ETSF availability.

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
INIT-01	1(3)	The MPS/Aura simulator shall be capable of selecting a desired version of the PDB at operator request during initialization.	5/17/2001	Differences between the Aqua and Aura PDB schema will require a revisit of this requirement when the Aura PDB is finalized.
INIT-01.01	1	During initialization, the MPS/Aura simulator shall provide the operator with the capability to select one version of the Aura PDB from among those available.		
INIT-01.02	1	During initialization, if the operator does not select a version of the Aura PDB, the MPS/Aura simulator will default to the most recent version available.		
INIT	TBD	Dependence upon Oracle as a database repository shall be removed.	3/7/2001 5/17/2001	NEW. The SIMSS baseline software is being modified to ingest PDB flat files directly during initialization. The completion date of this activity is TBD. TO BE REMOVED. The SIMSS PDB flat file capability will not be ready in time.
MDL-01	TBD	The MPS/Aura simulator shall provide a telemetry parameter modeling capability. The purpose of this capability is to simulate the behavior of a limited set of telemetry parameters.	3/7/2001	The modeling requirements (MDL-n) are Placeholders. The MPS/Aura simulator shall inherit the SIMSS Modeling capability when it achieves sufficient maturity. The completion date of that activity is TBD. The exact modeling requirements are also TBD.
MDL-02		The MPS/Aura simulator shall turn on and off selected modeling under operator control.		
MDL-03		The MPS/Aura simulator shall be capable of changing between static, table, or algorithm models under operator control		
MDL-04		The MPS/Aura simulator shall accept and execute modeling directives that enable or disable selected modeling.		
MDL-05		The MPS/Aura simulator shall accept and execute modeling directives that associate any telemetry parameter with any predefined model.		
MDL-06		The MPS/Aura simulator shall accept and execute modeling directives that change between static, table, or algorithm models.		
MDL-07		The MPS/Aura simulator shall provide the operator with an offline capability to access model functions and coefficients.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
MDL-08		The MPS/Aura simulator shall provide the operator with an offline capability to translate ASCII-formatted files containing static, table, and algorithm modeling information into a binary form readable by The MPS/Aura simulator.		
TLM-01	1	The MPS/Aura simulator shall be capable of switching between IP and serial modes of operation for command receipt and telemetry transmission.		
TLM-02	1	The MPS/Aura simulator shall provide the capability to transmit one stream of telemetry when in IP mode.		
TLM-03	1	The MPS/Aura simulator shall be capable of independently configuring telemetry and CLCW transmit when in IP mode.		
TLM-03.01	1	The MPS/Aura simulator shall be capable of transmitting packets containing CLCWs independently of telemetry transmission when in IP mode		
TLM-03.02	1	The MPS/Aura simulator shall default the CLCW packet transmission rate to 8 packets per second.		
TLM-03.03	2	The MPS/Aura simulator shall be capable of adjusting the frequency of CLCW packet transmission under operator control. The purpose of this requirement is to permit the CLCW transmission rate to match that of the telemetry transmission.		
TLM-03.04	1	The MPS/Aura simulator shall be capable of independently configuring IP mode telemetry and CLCW transmission to UDP MULTICAST mode when in IP mode.		
TLM-03.05	1	The MPS/Aura simulator shall be capable of independently configuring IP mode telemetry and CLCW transmission to any valid UDP MULTICAST IP address when in IP mode.		
TLM-03.06	1	The MPS/Aura simulator shall be capable of independently configuring IP mode telemetry and CLCW transmission to any valid UDP MULTICAST Port number when in IP mode.		
TLM-03.07	1	The MPS/Aura simulator shall be capable of independently configuring IP mode telemetry and CLCW transmission to any block length between one and 6000 bytes when in IP mode.		
TLM-03.08	1	The MPS/Aura simulator shall be capable of independently configuring IP mode telemetry transmission to variable block length when in IP mode.		
TLM-04	1	The MPS/Aura simulator shall be capable of transmitting one stream of CADUs when in serial mode.		
TLM-04.01	1	When in serial mode, the MPS/Aura simulator shall build S-band I-Channel CADUs as described in the Aura Spacecraft to Ground ICD.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
TLM-04.02	1	When in serial mode, the MPS/Aura simulator shall build and transmit I-Channel Fill CADUs as described in the Aura Spacecraft to Ground ICD when there is not enough telemetry data available to fill a CADU.		
TLM-05	1	The MPS/Aura simulator shall accept and execute operator directives that set the value of any telemetry parameter by mnemonic and by parameter ID.	5/17/2001	
TLM-05.01	1	The MPS/Aura simulator shall permit the operator to update telemetry parameter values in decimal, hex, and octal raw data numbers, and in Engineering Units.	5/17/2001	
TLM-05.02	1*	The MPS/Aura simulator shall be capable of accepting directives to set telemetry values using simple expressions (addition, subtraction, multiplication, etc.), trigonometric functions (sin, cos, etc.), Boolean expressions, the values or other telemetry parameters, and any combination thereof.	5/17/2001	
TLM-05.03	2	The MPS/Aura simulator shall be capable of limit checking operator-entered telemetry values and informing the operator if a value is too large to fit into the packet space. If a value is too large for the packet space it will be set to zero.	5/17/2001	
TLM-06	1*	The MPS/Aura simulator shall be capable of accessing telemetry parameters by mnemonic and by parameter ID.	5/17/2001	Parameter ID is also known as Last Received Value (LRV).
TLM-06.01	1	The MPS/Aura simulator shall use the PDB to define raw-data-to-EU and EU-to-raw-data conversions for telemetry parameters.	5/17/2001	
TLM-07	1	The MPS/Aura simulator shall accept and execute operator directives that request the value of any telemetry parameter for display.		
TLM-07.01	1	The MPS/Aura simulator shall be capable of displaying telemetry parameter values in decimal raw counts and in Engineering Units		
TLM-07.02	1	The MPS/Aura simulator shall be capable of displaying multiple telemetry parameter values in a GUI window.		
TLM-07.03	1	The MPS/Aura simulator shall be capable of displaying multiple iterations of a GUI window for display and update of telemetry parameters.		
TLM-08	1	The MPS/Aura simulator shall accept and execute operator directives that request the contents of any telemetry packet.		
TLM-09	1	The MPS/Aura simulator shall accept and execute operator directives that set the value of any location in the Aura-simulated spacecraft memory.	5/17/2001	
TLM-09.01	1	The MPS/Aura simulator shall accept and execute operator directives that request the value of any location or block of locations in simulated spacecraft memory.		
TLM-10	1	The MPS/Aura simulator shall set initial telemetry parameter values from information extracted from the Aura PDB.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
TLM-11	1	The MPS/Aura simulator shall accept and execute operator directives that result in changes to telemetry packet header values.		
TLM-12	TBD	The MPS/Aura simulator shall be capable of a <TBN> emulation of Solid State Recorder operation.	3/7/2001 5/17/2001 9/28/2001	PLACEHOLDER. Exact requirement is TBN. A low fidelity emulation may be achieved through the use of scenario scripts. REWRITTEN. SUPERCEDED.
	TBD	The MPS/Aura simulator shall be capable of a low-fidelity emulation of Solid State Recorder operation via scenario scripts.	9/28/2001	NEW.
TLM-13	1	The MPS/Aura simulator shall provide for the storage of housekeeping telemetry to be used as playback data.		
TLM-14	1	The MPS/Aura simulator shall use the information from the Aura PDB to generate and transmit telemetry packets.		
TLM-14.01	1	The MPS/Aura simulator shall be capable of creating CCSDS-format telemetry packets from information contained in the Aura PDB telemetry packet specification file.		
TLM-14.02	1	The MPS/Aura simulator shall provide the capability to generate and transmit telemetry packets with APIDs identical to the Aura spacecraft.		
TLM-14.03	1	The MPS/Aura simulator shall use the APID and secondary key fields of the Aura PDB packet definition file to identify unique packets.		
TLM-14.04	1	The MPS/Aura simulator shall generate a telemetry packet for each unique combination of APID and secondary key.		
TLM-14.05	1	The MPS/Aura simulator shall accept and execute operator directives to set the packet generation rate for any APID defined in the Aura PDB.		
TLM-14.06	1	The MPS/Aura simulator shall populate the telemetry packet primary header fields in the following list in accordance with information obtained from the Aura Spacecraft to Ground ICD and applicable CCSDS documents: Version Number, Type, Secondary Header Flag, APID, Sequence Flag, Sequence Count, and Packet Length.		
TLM-14.07	1	The MPS/Aura simulator shall place the secondary key into the telemetry packet at the offset specified by the Aura PDB telemetry packet specification file and shall use the number of bits specified by that file.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
TLM-14.08	1	The MPS/Aura simulator shall generate a telemetry packet secondary header in accordance with the secondary header type (SC, GIRD, or None for SUROM-TIE packets) implied by the contents of the packet type field of the Aura PDB telemetry packet specification file. For each secondary header type, the contents shall be as described in applicable sections of the Aura Spacecraft to Ground ICD.		
TLM-15	1	The MPS/Aura simulator shall insert simulated spacecraft time in the telemetry packet headers		
TLM-16	1	The MPS/Aura simulator shall maintain data values for all telemetry parameters defined in the PDB telemetry parameter specification file. These data values shall be available for display to the operator and for inclusion into telemetry packets.		
TLM-16.01	1	The MPS/Aura simulator shall be capable of inserting telemetry point values into packets using information from the Aura PDB telemetry description and telemetry parameter specification files.		
TLM-16.02	1	The MPS/Aura simulator shall use the APID and secondary key fields of the PDB telemetry parameter specification file to determine the correct packet for each telemetry parameter.		
TLM-17	1	The MPS/Aura simulator shall send out telemetry packets at specified intervals of spacecraft time. These specific intervals shall be as defined by the PDB for each APID and secondary key combination and shall be modifiable by the operator.		
TLM-18	1	The MPS/Aura simulator shall be capable of simulating spacecraft memory dumps. The MPS/Aura simulator shall build packets based on the contents of the simulated spacecraft memory.		
TLM-19	1*	The MPS/Aura simulator shall accept and execute directives that start and stop transmission of telemetry data.		
TLM-19.01	1*	The MPS/Aura simulator shall start transmission of telemetry and CLCW packets upon receipt of a start telemetry directive when in IP mode, unless startup of CLCW packet transmission is unlinked from telemetry startup. In that case only telemetry packet transmission shall be started.	3/7/2001	
TLM-19.02	1*	The MPS/Aura simulator shall stop transmission of telemetry and CLCW packets upon receipt of a stop telemetry directive when in IP mode, unless stopping of CLCW packet transmission is unlinked from telemetry stop. In that case only telemetry packet transmission shall be stopped.	3/7/2001	
TLM-19.03	1*	The MPS/Aura simulator shall be capable of starting and stopping the transmission of CLCW packets independently of telemetry transmission when in IP mode.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
TLM-19.04	1*	The MPS/Aura simulator shall be capable of maintaining a flag, under operator control, which, when set, shall link startup of CLCW packet transmission to telemetry transmission startup when in IP mode.	3/7/2001	
TLM-20	1	The MPS/Aura simulator shall accept and execute operator directives that start and stop logging of telemetry and CLCWs independently.		
TLM-21	1	The MPS/Aura simulator shall generate EDUs and EDOS data headers based on the User Datagram Protocol (UDP) format defined in the EDOS External ICD Data Format Control Document.		
TLM-22	1	The MPS/Aura simulator shall allow modification of any field within the EDOS data header.		
TLM-23	1	The MPS/Aura simulator shall provide the capability of transmitting the CLCW in the form of EDUs to EOC through EBnet.		
TLM-25	1	The MPS/Aura simulator shall provide for the storage of EDUs during the testing session for later transmission.		
TLM-26	1	The MPS/Aura simulator shall transmit EDUs on an as built basis.		
TLM-27	1	The MPS/Aura simulator shall provide the capability to transmit EDUs using the UDP protocol.		
TLM-28	1	The MPS/Aura simulator interface with the EOC shall comply with the telemetry interface formats and protocols specified in the EDOS to EGS Elements interface document		
TLM-29	1	The MPS/Aura simulator, when acting as a spacecraft, shall comply with the telemetry data formats and protocols specified in the TGT to EDOS interface document.		
TLM-30	1	The MPS/Aura simulator, when acting as an EPGS, shall comply with the telemetry data formats and protocols specified in applicable interface documents for the EPGS to EDOS interface.		
TLM-31	1	The MPS/Aura simulator shall provide the capability to accept Aura telemetry data by electronic transmission and by physical media.		
TLM-32	1	The MPS/Aura simulator shall be capable of transmitting the contents of a user provided file containing Aura telemetry data.		
TLM-34	1	The MPS/Aura simulator shall provide the capability to store up to 8MB of transmitted EDUs.		
TLM-35	1	The MPS/Aura simulator shall be capable of modifying multiple consecutive buffer locations via a single operator directive.		

Attachment G: Requirements Matrix

MPS/Aura Requirement	Release	MPS/Aura Requirement Description	Date of Insert/Update	Comments
TLM-36	1*	The MPS/Aura simulator shall be capable of a limited simulation of clock correlation telemetry. This simulation shall be limited to populating APID 1000 packets with simulated time and a simulated VCDU sequence counter value, as described in the Aura Spacecraft to EOS Ground System ICD	3/7/2001	
TLM-36.1	1	The MPS/Aura simulator shall maintain correlation between the simulated VCDU sequence counter transmitted in APID 1000 packets and in CLCW packets, when operating in IP mode.	3/7/2001 9/28/2001	Removed comment.
TLM-36.2	1	The MPS/Aura simulator shall maintain correlation between the simulated VCDU sequence counter transmitted in APID 1000 packets and the enclosing CADU, when operating in Serial mode.	3/7/2001 9/28/2001	Removed comment.
TLM-37	3	The MPS/Aura simulator shall be capable of transmitting telemetry packets over a 1553B interface, employing multiple Remote Terminal numbers and subaddresses.	3/7/2001 9/28/2001	See also requirement GEN-21. Implementation delayed due to delay in ETSF availability.
TLM	TBD	The MPS/Aura simulator shall be capable of reading telemetry packets from a disk file and interleaving them into the stream of simulator-generated packets.	9/28/2001	Proposed. To Be Removed.
TLM	TBD	The MPS/Aura simulator shall be capable of simulating an instrument memory dump of loaded data.	3/7/2001 9/28/2001	Proposed. The exact capability is TBN. To Be Removed.

The requirements defined as To Be Determined (TBD) and/or To Be Negotiated (TBN) are capabilities that may be added to the MPS/Aura simulator. The exact Level 4 requirements will be defined after negotiation of the desired capability.

Attachment H – System Limitations

H.1 MPS/Aura Release 3.0 Limitations

The following limitations apply to MPS/Aura Release 3.0. Some of these are Discrepancy Reports (DRs) against SIMSS baseline products and have been recorded in their DR repository.

Problem Description	Workaround
The Load Database window Version field will accept no more than 19 characters. A DR will be written.	Limit the PDB Version name to 19 characters when ingesting a PDB into Oracle.
The Scenario module File Selection window does not always show all of the files in the selected folder.	Click the Accept button of the File Selection window without selecting any file. Then type the scenario file name into the Filename field of the Scenario Control window, or copy and paste it from Windows Explorer.
The Save Project (Extended) and Restore From (Extended) options are intended for another application where a remote server runs simultaneously with the local application. The options are included with MPS/Aura so that only one version of the NeTT Core code needs to be maintained.	Avoid use of the Save Project (Extended) and Restore From (Extended) options.
The Generic Container Buffer display is limited to 1400 bytes of data (= 700 words, or 350 double words). A request for more data than that will result in a display of 1400 bytes of information. *This is SIMSS Defect # 102.*	To view data that is beyond byte 1400 of the buffer, set the offset to 1400, or as required to view the data.
If a container item name such as a telemetry mnemonic is entered into multiple displays of the *Display/Set Container Items…* window and updated in a higher numbered display, the update will not be reflected in lower numbered display(s).	Do not duplicate container item names.
When using SQL*Plus to select entries from the Oracle calcurve table via the conversion type field, *conv_type*, it is necessary to put a space after the type entry. e.g. “U_5D “, not “U_5D”.	Given at left.

Attachment I - Release History Summary Matrix

Attached is the MPS/Aura simulator release history summary matrix, updated to reflect the MPS/Aura Release 3.0 delivery. Modules inherited from the SIMSS baseline have the SIMSS Release Number, while the MPS/Aura modules EOSGS and SCAURA have the current MPS/Aura Release Number.

Release History Summary Matrix

System: **MPS/Aura**

Release Number		1.0	2.0	3.0 Beta	3.0									
Delivery Date		3/16/01	6/15/01	9/28/01	1/11/02									
Configuration Item	CI No.													
Core (Client)	1.1	4.0	4.1	5.0	6.0									
Core (Server)	1.2	4.0	4.1	5.0	6.0									
SCAURA (Client)	1.3	1.0	2.0	3.0	3.0									
SCAURA (Server)	1.4	1.0	2.0	3.0	3.0									
EOSGS (Client)	1.5	1.0	2.0	3.0	3.0									
EOSGS (Server)	1.6	1.0	2.0	3.0	3.0									
IP Input (Client)	1.7	4.0	4.1	5.0	6.0									
IP Input (Server)	1.8	4.0	4.1	5.0	6.0									
IP Output (Client)	1.9	4.0	4.1	5.0	6.0									
IP Output (Server)	2.0	4.0	4.1	5.0	6.0									
Logging (Client)	2.1	4.0	4.1	5.0	6.0									
Logging (Server)	2.2	4.0	4.1	5.0	6.0									

Delivery Date		3/16/01	6/15/01	9/28/01	1/11/02									
Configuration Item	CI No.													
Scenario (Client)	2.3	4.0	4.1	5.0	6.0									
Scenario (Server)	2.4	4.0	4.1	5.0	6.0									
Serial Input (Client)	2.5	4.0	4.1	5.0	6.0									
Serial Input (Server)	2.6	4.0	4.1	5.0	6.0									
Serial Output (Client)	2.7	4.0	4.1	5.0	6.0									
Serial Output (Server)	2.8	4.0	4.1	5.0	6.0									
TxFile (Client)	2.9	4.0	4.1	5.0	6.0									
TxFile (Server)	3.0	4.0	4.1	5.0	6.0									
vcProcessor (Client)[1]	3.1		4.1	5.0	6.0									
vcProcessor (Server)[1]	3.2		4.1	5.0	6.0									

[1] Added with Release 2.0

Attachment J - Delivery Details

J.1 Software

A complete listing of the MPS/Aura software file names is available upon request.

J.2 Hardware for MPS/Aura

The existing PCs on which the MPS/Aqua simulator runs may also be used to run the MPS/Aura simulator.

J.2.1 Hardware for MPS/Aura units in GSFC, Building 32

There are two CSOC-owned PCs presently installed in Building 32.

PC #1:

Qty	Common Name	Model [Serial No.]	Mfg	CSOC No.	Description
1	Computer	E-4200 001-343-8943	Gateway	C0060047	Intel Pentium II 400 Mhz w /512 Cache, 256 MB SDRAM PC100 6ns Micron, Matrox Millenium II 8MB AGP Video card, Toshiba 32x SCSI CD ROM Drive, Seagate 9.1 GB hard disk, IOMEGA 100 mb internal zip drive
1	Monitor	VX1100 811053233	Gateway	C0060041	21" Monitor
1	Mouse	Intellimouse 2570734-10000	Gateway		
1	Keyboard	Q9045A1837	Gateway		
1	Timing Card	PCIDCC20-P	Industrial Computer Source		PCI counter/timer card
2	Serial I/O Cards	97B1423 97B1424			Mfg name is FASTCOMESCC/P

PC #2:

Qty	Common Name	Model [Serial No.]	Mfg	CSOC No.	Description
1	Computer	E-4200 001-343-8944	Gateway	C0060050	Intel Pentium III 450 Mhz w /512 Cache, 256 MB SDRAM PC100 6ns Micron, Matrox Millenium II 8MB AGP Video card, Toshiba 32x SCSI CD ROM Drive, Seagate 9.1 GB hard disk, IOMEGA 100 mb internal zip drive
1	Monitor	VX1100 811053233	Gateway	C0060041	21" Monitor
1	Mouse	Intellimouse	Gateway		
1	Keyboard		Gateway		
2	Serial I/O Cards	97B1428 97B1429			Mfg name is FASTCOMESCC/P

J.2.2 Hardware for MPS/Aura at Denver

There are two CSOC-owned PCs currently installed at Denver.

PC #1:

Qty	Common Name	Model [Serial No.]	Mfg	CSOC No.	Description
1	Computer	E-4200 001-343-8946	Gateway	C0060052	Intel Pentium III 450 Mhz w /512 Cache, 256 MB SDRAM PC100 6ns Micron, Matrox Millenium II 8MB AGP Video card, Toshiba 32x SCSI CD ROM Drive, Seagate 9.1 GB hard disk, IOMEGA 100 mb internal zip drive
1	Monitor	VX1100 811053230	Gateway	C0060043	21" Monitor
1	Mouse	Intellimouse	Gateway		
1	Keyboard		Gateway		

PC #2:

Qty	**Common Name**	**Model [Serial No.]**	**Mfg**	**CSOC No.**	**Description**
1	Computer	E-4200 001-343-8945	Gateway	C0060048	Intel Pentium III 450 Mhz w /512 Cache, 256 MB SDRAM PC100 6ns Micron, Matrox Millenium II 8MB AGP Video card, Toshiba 32x SCSI CD ROM Drive, Seagate 9.1 GB hard disk, IOMEGA 100 mb internal zip drive
1	Monitor	VX1100 811053463	Gateway	C0060044	21" Monitor
1	Mouse	Intellimouse	Gateway		
1	Keyboard		Gateway		

Attachment K - Documentation References

The following documents have been employed as the main sources for direction and information in producing Release 3.0 of the MPS/Aura simulator.

Document	Location[*]
Earth Observing System (EOS) Common Spacecraft Program Interface Control Document between the EOS Aura Spacecraft and the EOS Ground System, Dated April 15, 2000, Document No. D27515, Preliminary (more commonly known as "The Space to Ground ICD")	1
Data Format Control Document for the Earth Observing System (EOS) Mission Operations Segment (EMOS) Project Database Volume 1: Aura Users Revision - TBD, undated	4
TRW, EOS Chemistry Spacecraft Flight Software Requirements Specification, ES-SDA-005, Rev. A, dated 23 August, 2000	1
TRW, EOS Aura Spacecraft Flight Software User's Guide, No.: D31189, version dated 15 September, 2000	1
TRW, EOS Aura Command Allocation Document, No.: D31174, dated July 27, 2000	1
TRW, EOS Aura Telemetry Allocation Document, No.: D31175, dated September 22, 2000	1
TRW, EOS PM-1 Spacecraft Equipment Specification for Transponder Interface Electronics, No.: EQ4-4957, latest version dated 11 February, 1999	1
TRW, Interface Control Document Between the Earth Observing System (EOS) Data and Operations System (EDOS) and the EOS Ground System (EGS) Elements CDRL B301	2
TRW, Interface Control document for the High Resolution Dynamics Limb Sounder (HIRDLS), Rev. B No.: D26477, dated July 31, 2000	1
TRW, Interface Control Document for the Microwave Limb Sounder (MLS), Rev B No.: D26475, dated July 31, 2000	1
TRW, Interface Control Document for the Ozone Monitoring Instrument (OMI) System, Rev. A No.: D26478, dated April 29, 2001	1
TRW, Interface Control Document for the Tropospheric Emission Spectrometer (TES), Rev. C No.: D26476, dated August 30, 2001	1
TRW, TES Command and Telemetry Handbook, Version 4.0 No.: D15388, dated Sep. 14, 2001	1
Consultative Committee for Space Data Systems, CCSDS 102.0-B-4: Packet Telemetry Blue Book, Issue 4, Nov. 1995	3
--, CCSDS 202.1-B-1: Telecommand Part 2.1 – Command Operations Procedures Blue Book, Issue 1, Oct. 1995	3
NASA, GSFC, Earth Observing System Data and Information System (EOSDIS) Test System (ETS) Level 4 Requirements for the MPS/Aura. Version listed in Attachment G.	This package.

[*] See Location Legend

Location Legend:

Number	Designation
1	TRW web server (URL not listed for security reasons)
2	http://esdis-it.gsfc.nasa.gov:8080/servlet/DOCcat?nCatType=ICD
3	http://ccsds.org/publications.html
4	hardcopy

Attachment L — Mission Systems Configuration Management Form

This attachment contains the completed Mission Systems Configuration Management (MSCM) form for the delivery of MPS/Aura Release 3.0.

Mission Systems Configuration Management Form

1. ORIGINATOR	2. ORGANIZATION	3. PHONE	4. E-MAIL ADDRESS
Dave Green	CSC	301-805-3420	dsgreen@csc.com

5. ELEMENT	6. INSTALLATION PRIORITY	7. TRACKING NUMBER
ETS (MPS/Aura)	Routine	(Assigned by CM Office)

8. SOURCE CHANGE REQUEST(S):

ETS delivery of MPS for EOS Aura (MPS/Aura)

9. APPROVALS

Element Manager	____________________	____/____/____
Flight Ops Director	____________________	____/____/____
Operations Manager	____________________	____/____/____

10. DELIVERED SYSTEM (Check all that apply)

	Name	Version	Media Identification	Identification Date
□ Hardware				
☒ Software	MPS/Aura	R3.0	CD-ROM	01/11/02
□ Database				
☒ Documentation:				
	MPS/Aura delivery package	N/A	3.5 " Diskette	01/11/02
	MPS/Aura Release 3.0 User's Guide	R3.0	http://esdis-it.gsfc.nasa.gov/ETS/etsdoc.html	TBS
□ Other				

11. CHANGE DESCRIPTION

Release 3.0 of MPS/Aura

12. ATTACHMENT(S): Check if YES ☒

Description: MPS/Aura Release 3.0 delivery package (cover letter with attachments) dated 01/11/02

13. CM OFFICE USE

	Location (Bldg/Room)	Slot location(s)
Hardware	____/____	
Media	____/____	
Documentation	____/____	

Installation date ____/____/____ CM Office Signature ____________________

Form MSCM (970327)